



Redireccionamiento por el navegador

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de Servlet. En el video pasado nos quedamos con este problema, en el cual nosotros si hacíamos un F5 estábamos enviando de nuevo el formulario y entonces estábamos registrando muchas veces nuestra empresa Alura en este caso.

[00:19] ¿Entonces cómo hacemos para resolver este problema? Vean lo siguiente, nosotros tenemos en este momento este esquema, nosotros tenemos una request que es un post que va a nuestro Servlet de nueva empresa y a través de ahí usamos dispatcher para enviar esa request y devolver una respuesta.

[00:38] La solución a nuestro problema sería algo como el estilo, tener una request a nuestro servidor que registra la empresa en nuestra base de datos y luego nosotros hacer un redireccionamiento, o sea decirle a nuestro navegador: “navegador, después de que yo termine con esto, podés ir hacia el siguiente sitio”.

[01:00] Entonces le hacemos un redireccionamiento client side o sea del lado del cliente, y entonces el navegador hace esa última request hacia nuestro Servlet de listar empresas que de la base de datos obtiene los datos y muestra nuestro JSP.

[01:20] Entonces, después cuando nosotros estamos ya acá que recibimos esta última respuesta con toda la lista, cuando hacemos F5 nosotros enviamos est

request de nuevo, entonces no estamos registrando de nuevo una nueva empresa, sino que estamos listando las empresas de nuevo.

[01:36] Entonces eso solucionará nuestros problemas. Lo que tenemos acá es un redireccionamiento server side, que ocurre todo dentro del servidor y esto de acá es un redireccionamiento client side, o sea del lado del cliente, del lado del navegador, nosotros hacemos un request que lleva unos parámetros, y luego hacemos, le decimos al navegador: “andá a esta otra web, a este otro Servlet”.

[02:05] Entonces el navegador hace una request a ese otro Servlet. Vamos a ver cómo es eso en el código. Primero vamos a comentar todo esto que es la parte del RequestDispatcher. “Ctrl + Shift + C”. Seleccionamos las líneas que queremos comentar y ahí hemos comentado.

[02:30] Nosotros lo que tenemos que utilizar es nuestro response, en este caso, punto, y acá nosotros tenemos que hacer un redirect, sendRedirect. Utilizamos el ese sendRedirect y tenemos que decirle hacia dónde tiene que redirigirse nuestro navegador. Entonces le tenemos que decir ahora, en esta respuesta le decimos: “Tenés que redirigirte hacia este Servlet”.

[02:54] Entonces nosotros acá vamos a colocar “listaEmpresas”, que ese es nuestro Servlet. Se coloca “listaEmpresas” sin la barra, eso es muy importante. Un dato curioso: este redireccionamiento client side surgió cuando antiguamente las personas querían acceder a un documento y ese documento había sido eliminado o había sido movido hacia otra dirección.

[03:24] Entonces este redirectamente client side surgió dentro de lo que es HTTP para poder resolver ese problema, para poder decir: “no tengo el documento en este lugar donde estás buscando, pero podés ir a buscarlo a esta otra dirección”. Entonces con eso resolvían esos problemas. Bien.

[03:44] Una vez hecho este redirect, vamos a hacer un "Ctrl + S", vamos a hacer un run de nuevo acá y vamos a probar con nuestro formulario. Vamos a dar un

F5, voy a colocar Alura en este caso, vamos a enviar y vemos que se registró. Veán lo siguiente, me olvidé de mostrar una cosa.

[04:07] Aprieten F12 para poder ir a la consola de desarrollador de Google Chrome o Firefox, si están en Firefox, posiblemente ustedes vean ese elements en este momento, entonces vayan a la pestaña de network. Y vamos a hacer lo siguiente, voy a volver para atrás. Una vez que estoy acá, quiero eliminar, hacer un clear porque no quiero que vean esta request aun.

[04:33] Entonces dentro de nuestro formulario voy a enviar la request de, por ejemplo, Caelum, bien, vamos hacer clic en enviar. Y bueno, se registró Caelum y vean lo siguiente, vean lo que ocurrió. Nosotros tenemos acá dos requests, ocurrió primero ese request nuevaEmpresa, y después ocurrió esa listaEmpresas, entonces hubo dos requests.

[04:58] En la primera, nosotros tenemos un post y un status code que es 302. ¿Qué significa ese status code? Bueno, yo voy a cerrar acá y vean lo siguiente, un status code, como ya les había dicho anteriormente, es una cosa que tiene que estar sí o sí, es una de las cosas más importantes dentro de lo que es una respuesta de nuestro servidor.

[05:22] Entonces el servidor sí o sí tiene que dar una respuesta de status code, por lo menos, entonces ese 302, nosotros ya vimos que el 200 significaba que estaba todo bien, el 404, por ejemplo, decía que no encontrábamos, no estaba encontrando el documento que se estaba necesitando en ese lugar y el 302 significa un redirect.

[05:44] Pueden ver acá en el tipo que se llama redirect. Entonces, ese nuevaEmpresa tiene un post y tiene un direccionamiento ¿hacia dónde? Dentro de los headers podemos ver que está ese location y dice “lista empresas”. Con esto el navegador ya sabe que tiene que hacer un redireccionamiento 302 hacia lista empresas y por eso que llamó a ese listaEmpresas después.

[06:07] Que devolvió, fue un método GET y obtuvo un 200 que está todo bien. Bien. Ahora vean lo siguiente, vimos desde el formulario hacia nuestra lista como es las requests, que ocurren dos requests. Y vean lo que ocurre ahora. Si yo hago un F5, vean que ahora solamente se hizo una request, se hizo un listaEmpresas. ¿Por qué?

[06:35] Porque está ocurriendo esto de acá. Nosotros ya hicimos una request desde nuestro formulario. Recibimos un redirect o una redirección, hicimos una segunda request. Y entonces, cuando nosotros hacemos F5, nos mantenemos dentro de este loop. Entonces no estamos enviando una nueva empresa para registrarse.

[06:53] Entonces por eso podemos ver eso y además bueno, vean que listaEmpresas de acá arriba está funcionando mientras que yo comente todo ese requestDispatcher, entonces, y aún así podemos ir hacia nuestra lista de empresas, entonces significa que ese redireccionamiento funcionó. Sin embargo, ahora tenemos otro problema.

[07:16] Cuando nosotros enviamos un formulario, voy a cerrar acá, voy a dar un F5, vean que voy a colocar Google OK, cuando veamos un formulario, vean que ahora ya no está apareciendo más el mensaje que aparecía arriba, ese de que empresa Google, registrada en este caso, no está apareciendo, ¿por qué no está apareciendo?

[07:34] Veán lo siguiente. En nuestra lista de empresas acá., dentro de esa lista de empresas nosotros teníamos un if, que si en la empresa no era vacío, entonces mostrábamos ese mensaje de empresa Google registrada en este caso.k Bien. ¿Por qué no está existiendo más esa empresa?

[07:58] Veán lo siguiente, en nuestro Servlet de nueva empresa registrada nosotros teníamos una empresa que habíamos creado, que le habíamos colocado un nombre, una fecha, y esa empresa, voy a copiar acá el

RequestDispatcher que habíamos hecho, y esa empresa nosotros la colocábamos dentro de nuestra request.

[08:23] Entonces la empresa que obteníamos la colocamos dentro de nuestra request y ese recuesta, nosotros lo íbamos enviando entre cada uno de los Servlets, lo enviamos a los dispatchers. Entonces ese request que colocábamos la empresa iba acá y esa empresa iba acá.

[08:42] La cuestión es que con esta nueva metodología, ahora nosotros nuestra request, si nosotros colocamos nuestra empresa en esa request, esa empresa, esa request solamente vive durante el tiempo que dura la request, o sea, parece bobo lo que estoy diciendo, pero en realidad esa empresa tiene un scope que solamente vive hasta que nosotros enviamos una respuesta.

[09:10] Entonces tenemos una request, colocamos una empresa y cuando enviamos la respuesta, luego de eso ya esa empresa, ese objeto empresa que nosotros creamos en ese momento, deja de existir en nuestro servidor. Claro que nosotros lo guardamos a esa empresa dentro de nuestra base de datos, en eso no hay problema.

[09:31] Pero entiendan lo siguiente, esa request vivió acá y murió acá, entonces cualquier cosa que le pusimos ya no existe más, entonces cuando llegó esta segunda request ya no existía más esa empresa, ese objeto empresa nuevo que habíamos creado como objeto, sí existe de nuevo dentro de la base de datos.

[09:52] Entonces cuando no encuentra una empresa, esa empresa, objeto empresa no va a mostrar el mensaje como vemos acá. Si la respuesta empresa fuera vacía, si fuera vacía no se ejecuta esto. Cuando no es vacía, cuando tiene alguna cosa, entonces se ejecuta.

[10:13] Entonces nosotros resolvimos un problema, pero creamos otro, entonces ¿cómo hacemos para resolver ese problema? Desde ya les adelanto que en este video no lo vamos a resolver porque necesitamos conocimiento de algunas otras técnicas, pero sí les puedo decir que es un problema de scope.

[10:33] Entonces es una cuestión de alcance. Cuando nosotros teníamos esto de acá, esa variable, ese objeto empresa solamente existió durante ese alcance, ese scope, acá no existe más. Entonces necesitamos alguna cosa que tenga un scope, un alcance que no sea de nuestra request, algo que tenga un scope mayor bien, y que no viva durante solamente la vida útil de nuestra request.

[11:01] Entonces eso ya lo vamos a ver en otro video. Hasta ahora, simplemente hagan los ejercicios y nos vemos el próximo video.